

MULTI-TASK PROCESSING METHOD AND PROCESSOR

Publication number: JP2002189604 (A)

Publication date: 2002-07-05

Inventor(s): TAKAHASHI KATSUMI +

Applicant(s): TOSHIBA CORP +

Classification:

- International: G06F9/46; G06F9/46; (IPC1-7): G06F9/46

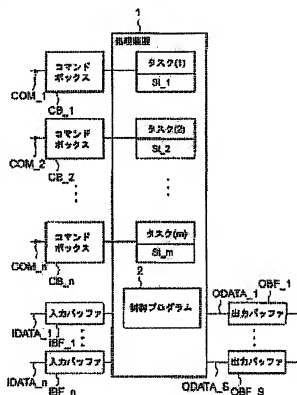
- European:

Application number: JP20000387405 20001220

Priority number(s): JP20000387405 20001220

Abstract of JP 2002189604 (A)

PROBLEM TO BE SOLVED: To provide a multi-task processing method capable of multi-task processing without impairing response to a command and a processor using the same. **SOLUTION:** When switching takes place between tasks to be processed in order to execute a plurality of different processes by multi-task processing, the next task to be processed is determined based on a plurality of criteria on the respective tasks. At least one of the plurality of tasks is a task that receives a receivable command previously determined for processing the task and processes the command, that this task has received the command is set as one of the plurality of criteria, and the task having received the command is preferentially determined as the next task to be processed.



Data supplied from the *espacenet* database — Worldwide

(51) Int.Cl. ⁷	機別記号	F I	データベース(参考)
G 0 6 F 9/46	3 4 0	C 0 6 F 9/46	3 4 0 B 5 B 0 9 8

審査請求 未請求 請求項の数6 O L (全 7 頁)

(21) 出願番号 特願2000-387405(P2000-387405)

(22) 出願日 平成12年12月20日(2000.12.20)

(71) 出願人 00003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72) 発明者 高橋 克己

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

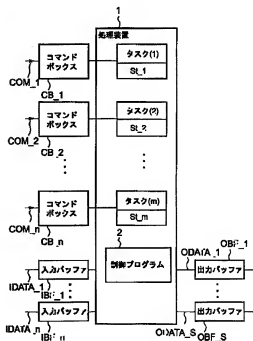
Fターム(参考) 5B09B C001 GA04 GC03 GD02 GD14

(54) 【発明の名称】 マルチタスク処理方法および処理装置

(57) 【要約】

【課題】 コマンドの応答性を損わずにマルチタスク処理を可能にするマルチタスク処理方法およびそれを用いた処理装置を提供する。

【解決手段】 複数の異なる処理をマルチタスク処理にて実行するため、処理対象のタスクを切り替えるにあたって、次の処理対象のタスクを各タスクの複数の判断基準に基づき決定するものであって、複数のタスクのうちの少なくとも1つは、そのタスクの処理に予め定められた受け付け可能なコマンドを受け付けて、そのコマンドを処理するタスクであって、このタスクが前記コマンドを受け付けていることを前記複数の判断基準のうちの1つとし、前記コマンドを受け付けたタスクを優先的に次の処理対象のタスクとして決定する。



【特許請求の範囲】

【請求項1】 複数の異なる処理をマルチタスク処理にて実行するとともに、処理対象のタスクの切り替えにあたっては、次の処理対象のタスクを複数の判断基準に基づき決定するようにしたマルチタスク処理方法において、

複数のタスクのうちの少なくとも1つは、そのタスクの処理において予め定められたコマンドを受け付けて、そのコマンドを処理するタスクであって、このタスクが前記コマンドを受け付けていることを前記複数の判断基準のうちの1つとし、前記コマンドを受け付けたタスクを優先的に次の処理対象のタスクとして決定することを特徴とするマルチタスク処理方法。

【請求項2】 前記コマンドを受け付けたタスクについて、前記複数の判断基準のうちの他の判断基準に基づきそれぞれの優先度を算出し、この算出された優先度に基づき次の処理対象のタスクとして決定することを特徴とする請求項1記載のマルチタスク処理方法。

【請求項3】 前記コマンドを受け付けているタスクの有無を無視して、まず、全てのタスクについて、前記複数の判断基準のうちの他の判断基準に基づきそれぞれ優先度を算出し、この算出された優先度が最も高いものが複数あるとき、その複数のタスクのうち前記コマンドを受け付けたタスクを次の処理対象のタスクとして決定することを特徴とする請求項1記載のマルチタスク処理方法。

【請求項4】 複数の異なる処理をマルチタスク処理にて実行するとともに、処理対象のタスクの切り替えにあたっては、次の処理対象のタスクを複数の判断基準に基づき決定するようにしたマルチタスク処理方法において、

複数のタスクのうちの少なくとも1つは、そのタスクの処理において予め定められたコマンドを受け付けて、そのコマンドを処理するタスクであって、このタスクが受け付けているコマンドの有無およびコマンドの数およびコマンドの種類のうち少なくとも1つを前記複数の判断基準のうちの1つとして各タスクのそれぞれの優先度を算出して、この算出された優先度に基づき次の処理対象のタスクを決定することを特徴とするマルチタスク処理方法。

【請求項5】 複数の異なる処理をマルチタスク処理にて実行する処理装置において、

処理対象のタスクを切り替えるにあたって、次の処理対象のタスクを複数の判断基準に基づき決定する決定手段を具備し、

複数のタスクのうちの少なくとも1つは、そのタスクの処理において予め定められたコマンドを受け付けて、そのコマンドを処理するタスクであって、前記決定手段は、このタスクが前記コマンドを受け付けていることを前記複数の判断基準のうちの1つとし、前記コマンドを

受け付けたタスクを優先的に次の処理対象のタスクとして決定することを特徴とする処理装置。

【請求項6】 複数の異なる処理をマルチタスク処理にて実行する処理装置において、

処理対象のタスクを切り替えるにあたって、複数の判断基準に基づきそれぞれの優先度を算出する算出手段と、この算出手段で算出された優先度に基づき次の処理対象のタスクを決定する決定手段とを具備し、

複数のタスクのうちの少なくとも1つは、そのタスクの処理において予め定められたコマンドを受け付けて、そのコマンドを処理するタスクであって、前記算出手段は、このタスクが受け付けているコマンドの有無およびコマンドの数およびコマンドの種類のうち少なくとも1つを前記複数の判断基準のうちの1つとして各タスクのそれぞれの優先度を算出することを特徴とする処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マルチタスク処理方法およびそれをを用いた処理装置に関する。

【0002】

【従来の技術】複数のタスクを単一の処理装置（プロセッサ）で処理する場合、時間分割して処理することが行われている。ある時刻には1個のタスクが実行されており、このタスクの実行が中断されると、制御プログラムに制御が移され、制御プログラムは次に実行するタスクを決定する。従来の方式では、制御プログラムは、実行可能なタスクの中で、各タスクの優先順位を示す、あらかじめ数値化された値の大小に基づいて、次に実行するタスクを決定する。詳細な決定アルゴリズムは幾通りかあるが、代表的なものは次の通りである。実行可能なタスクの中で、最も優先度の高いものが1個ある場合は、それが選択される。最も優先度の高いものが2個以上ある場合には、それらがサイクリックに順番に実行されるように決定される。

【0003】優先順位を示す数値は、タスクの実行中に固定であるものや変化するものがある。変化するものは、タスク自身が値を変更するのが普通の方法である。つまりプログラムの実行位置に応じて、高い優先度が必要になった場合には優先度を上げ、低い優先度でも良くなった場合には優先度を下げるのである。

【0004】ところが、インターネットやデジタル放送などで音楽や映像などのデータを受信しながらリアルタイムに再生する（ストリーミング）際には、連続的に入力されてくる音声データなどストリーミング・データで処理するプログラムの実行中に、あるタスクが外部から、例えば、処理中止のコマンド（「STOP」コマンド）やビットレート等のパラメータ変更のコマンド等を受けた場合、従来では、優先度を決定する際、コマンドの有無が全く考慮されていないため、コマンドを受け

たタスクの実行が遅れ、その結果、コマンドの応答性が悪くなるという問題点があった。

【0005】

【発明が解決しようとする課題】従来のマルチタスク処理は、固定の優先順位や、プログラマー自身が設定する優先度に基づいて処理対象のタスクの切り換えを制御するものであり、タスクが外部から所定のコマンドを受け付けた際に、そのタスクの優先度が低く、そのコマンドの処理が実行できずに、コマンドの応答性が悪いという問題点があった。

【0006】そこで、本発明は、上記問題点に鑑み、コマンドの応答性を損なわずにマルチタスク処理を可能にするマルチタスク処理方法およびそれを用いた処理装置を提供することを目的とする。

【0007】

【課題を解決するための手段】本発明は、複数の異なる処理をマルチタスク処理にて実行するとともに、処理対象のタスクの切り替えにあたって、次の処理対象のタスクを複数の判断基準に基づき決定するものであって、複数のタスクのうちの少なくとも1つは、そのタスクの処理において予め定められたコマンドを受け付けて、そのコマンドを処理するタスクであって、このタスクが前記コマンドを受け付けていることを前記複数の判断基準のうちの1つとし、前記コマンドを受け付けたタスクを優先的に次の処理対象のタスクとして決定することにより、コマンドの応答性を損なわずにマルチタスク処理を可能にする。

【0008】好ましくは、前記コマンドを受け付けたタスクについて、前記複数の判断基準のうちの他の判断基準に基づきそれぞれの優先度を算出し、この算出された優先度に基づき次の処理対象のタスクとして決定する。

【0009】また、好ましくは、前記コマンドを受け付けているタスクの有無を無視して、まず、全てのタスクについて、前記複数の判断基準のうちの他の判断基準に基づきそれぞれ優先度を算出し、この算出された優先度が最も高いものが複数あるとき、その複数のタスクのうち前記コマンドを受け付けたタスクを次の処理対象のタスクとして決定する。

【0010】また、本発明は、複数の異なる処理をマルチタスク処理にて実行するとともに、処理対象のタスクの切り替えにあたって、次の処理対象のタスクを複数の判断基準に基づき決定するものであって、複数のタスクのうちの少なくとも1つは、そのタスクの処理において予め定められたコマンドを受け付けて、そのコマンドを処理するタスクであって、このタスクが受け付けているコマンドの有無およびコマンドの数およびコマンドの種類の中の少なくとも1つを前記複数の判断基準のうちの1つとして各タスクのそれぞれの優先度を算出して、この優先度に基づき次の処理対象のタスクを決定することにより、コマンドの応答性を損なわずにマルチタスク

処理を可能にする。

【0011】

【発明の実施の形態】以下、本発明の実施形態について図面を参照して説明する。

【0012】図1は、本実施形態に係るデータ処理装置（以下、簡単に処理装置と呼ぶ）1の要部の構成を概念的に示したものである。処理対象のデータが複数（ここでは、例えばn系列）あり（入力データIDATA₁～IDATA_n）、それらはそれぞれ入力バッファIBF₁～IBF_nに入力される。これら入力データは、例えば、複数チャネルのオーディオ信号やビデオ信号などである。これらのデータは、マクロ的にはほぼ一定のレートで入力されることが多いが、ミクロ的には入力バッファに転送する側の性質に依存するものであり、バースト的に入力される場合もあろう。

【0013】図1における処理装置1において、各入力データに対する処理が遅れると、入力バッファがオーバーフローしてしまうため、それが起こらないように処理しなければならない。

【0014】処理装置1は、ハードウェア的には単一のプロセッサで構成されており、この上で複数（ここでは、例えばm個）のタスク（タスク（1）～タスク（m））が動作するものとする。

【0015】タスク（1）～タスク（m）は、入力バッファIBF₁～IBF_nに格納されているデータを逐次処理していくための処理プログラムである。

【0016】入力バッファIBF₁～IBF_nに格納されている入力データIDATA₁～IDATA_nは、タスク（1）～タスク（m）で処理されて、複数（ここでは、例えばs系列）の出力データODATA₁～ODATA_sとして出力され、それぞれ出力バッファOBF₁～OBF_sに格納される。なお、出力バッファがそのまま入力バッファに対応することもあるが、ここでは、説明の簡単のため、それぞれ別個に設けられているものとする。なお、m=nおよびまたはn=sであってもよい。

【0017】また、タスク（1）～（m）のそれぞれには、外部からのコマンドCOM₁～COM_nを受け取るために、インタフェース（図4では、コマンドボックスCB₁～CB_n）が接続されている。なお、ここでは、全てのタスク（1）～（m）にコマンドボックスCB₁～CB_nが接続されているが、少なくとも1個のタスクにコマンドボックスが接続されていばよい。

【0018】コマンドボックスCB₁～CB_nは、例えば、FIFO (First In First Out) メモリ、RAMであってもよい。

【0019】また、マルチタスク処理実行中に各コマンドボックスCB₁～CB_nに入力される（書き込まれる）コマンドとしては、ユーザにより入力されるコン

ドであったり、現在実行中のアプリケーションから入力されるものもあるが、具体的には、例えば、処理中止のコマンド（「STOP」コマンド）やビットレート等のパラメータ変更のコマンド、出力データを調整するためのリアルタイム性が要求されるコマンド（例えば、出力データが音声の場合にその音量を大きくしたり小さくしたりするためのコマンド）等がある。

【0020】図1の処理装置1は単一のプロセッサで構成されているので、同時には1個のタスクしか実行されず、 m 個のタスクは時間分割されて実行されることになる。なお、ここでは、単一のプロセッサで構成された処理装置の場合を例にとり説明するが、一般的には、 m より少ない複数のプロセッサで構成された処理装置でも同様である。

【0021】各タスクへの時間割り当てを決めるのは、制御プログラム2である。タスクの時間割り当ては次のようにして行われる。ある時刻にタスク(k) ($1 \leq k \leq m$ の整数)が実行されているとする。

【0022】タスク(k)はある条件に従って処理が中断される。この条件は、従来の一般的なマルチタスク制御方式においては、タスク(k)の実行が始まってから一定時間が経過したときである。すなわち、時間を固定長のタイム・スロットに分割し、各タイム・スロット毎にタスクを割り当てる方式である。

【0023】本発明においても、このような方式を用いることも可能である。また、この方式を用いない場合としては、タスク(k)が自ら実行を中断する方式が考えられる。後者の方式による利点は、自らの都合で中断を行えるため、処理上区切りがよいところで中断することにより、タスク切り替え時の時間が短くなり、効率がよいことである。すなわち逆に言えば、前者のように強制的に中断される方式では、後で再開するときに必要な中間データが大きくなり、中断時にそれらを保存したり、再開時にそれらを復元したりするのにより多くの時間を費やすことになる。後者の方式における処理上の区切りとは、例えば、入力データが圧縮符号化されたオーディオ信号の場合には、フレームの区切りとすると都合がよい。

【0024】本実施形態では、前者の予め定められた時間経過したときタスクの処理を中断させてタスクを切り替える方式であっても、後者のタスク自身が例えばフレームの区切りをきっかけに処理を中断したときタスクを切り換える方式であっても適用可能である。

【0025】以下、図2～図4に示すフローチャートを参照して、制御プログラム2のマルチタスク制御処理動作について説明する。

【0026】タスク(k)が処理装置のプロセッサにより実行され、その実行が中断されると、制御プログラム2に処理が移される。制御プログラム2は、OS（オペレーションシステム）が存在するシステムの場合には、

通常、OSの一部である。制御プログラム2は、コマンドの存在や、入出力系統の現状等の複数の判断基準に基づき次に実行するタスクを決定するものである。

【0027】従来の制御プログラムにおいては、各タスクに優先度を示す数値が設定されており、実行可能なタスクの中で、それらの数値の最も大きいものが次に実行されるタスクとして選択される。ここで、実行可能とは、図1のような処理系の場合には、各タスクに対応する処理の入出力系統において、その入力バッファが空でなく、かつ対応する出力バッファがフルでないことである。

【0028】しかしながら、図1に示す処理系では、各タスクにあらかじめ優先度をつけておくことや、タスク実行時に自分自身の優先度を設定しておくことは望ましくない。なぜなら、ここでの優先度のあるべき姿は、入力バッファがオーバーフローしないことや、出力バッファが空にならないことであり、これらの条件は当該タスクが実行していない場合にも時々刻々と変化するからである。すなわち優先度を表す数値が時々刻々と変化するようになっている必要がある。

【0029】そこで、図1では、時々刻々と変化する優先度を実現するために、各タスク(1)～タスク(m)のそれぞれにステータス関数 $st_1 \sim st_m$ を対応付け、制御プログラム2が次に実行すべきタスクを選択する時点において、各タスク(1)～タスク(m)のそれぞれに対応したステータス関数 $st_1 \sim st_m$ を呼び、各ステータス関数により、例えば、後述する算出方法にて入出力系統の現状に基づく優先度を算出した結果を取得する。

【0030】また、ストリーミングにおけるマルチタスク処理では、外部からコマンドを受け取ることもあり、そのような図1に示したような処理系では、従来の方式は好ましくない。なぜなら、コマンドボックスにコマンドが書き込まれているタスクとそうでないタスクとがある場合、その優先度を決める際に、コマンドの有無が全く考慮されないからである。そのため、外部から入力されたコマンドに対するレスポンスが遅くなる可能性が生じてしまう。

【0031】そこで、優先度を決める際に、それぞれのタスクに対応したコマンドボックスのコマンドの存在を考慮することにする。コマンドの存在をどのようにタスクの優先度に反映するするかは、処理装置1の構成に依存し、無数にその方法が存在する。その全てを個々に列挙することは不可能であるので、その一部について説明する。なお、以下に説明する方法を適宜組み合わせることも可能である。

【0032】まず、図2を参照して、制御プログラム2のマルチタスク制御処理（処理A）について説明する。

【0033】図2に示す処理Aは、次の処理対象としてのタスクを決定する際の判断基準としてのコマンド存在

が、他の判断基準より重要度が高い場合の方法である。すなわち、制御プログラム2は、各タスクのコマンドボックス内にコマンドが書き込まれているか否か(コマンドを受けているか否か)をチェックする(ステップS1)。コマンドボックスにコマンドが書き込まれているタスクが複数ある場合は、その複数のタスクのそれぞれの優先度をコマンド以外の判断基準に基づき算出する(ステップS2)。制御プログラムは、この複数のタスクの優先度から、次に実行すべきタスクとして、最も優先度の高いタスクを選択する。ここでは、その選択されたタスクをタスク(k)とする(ステップS4)。制御プログラムは、タスクの切り換え時に、プロセッサに、この選択されたタスク(k)を次の処理対象として実行させる。なお、コマンドボックスにコマンドが書き込まれているタスクが1つの場合は、そのタスクを次に実行すべきタスク(k)として選択する(ステップS4)。コマンドボックスにコマンドの書き込まれたタスクが存在しないときは、全てのタスクについて優先度を算出して、この算出した優先度が最も高いタスクを次に実行すべきタスク(k)として選択する(ステップS4)。

【0034】なお、ステップS2、ステップS3では、各タスクにあらかじめ付けられた優先度や、タスク実行時に自分自身で設定された優先度を用いてもよいが、好ましくは、入力バッファIBF₁～IBF_n、出力バッファOBF₁～OBF_sの入出力系統の現状を判断基準として、入力バッファがオーバーフローしないことや、出力バッファが空にならないことを条件に、タスク切替時に優先度をその都度算出することが望ましい。ここで、そのための優先度の算出方法の一例を説明する。すなわち、時々刻々と変化する優先度を実現するために、各タスク(1)～タスク(m)のそれぞれにステータス関数St₁～St_mを対応付け、制御プログラム2が、各タスク(1)～タスク(m)のそれぞれに対応したステータス関数St₁～St_mを呼び、各ステータス関数により、例えば、後述する算出方法にて優先度を算出した結果を取得する。

【0035】各ステータス関数St₁～St_mは、それらに対応したタスクの入出力系統の現状を評価し、それに応じた優先度を返すように作られている。また各タスクが実行可能か実行不可能かの情報も返すようにすることも可能となっている。ここで実行不可能とは、例えば入力バッファ占有量が空か、出力バッファ占有量がフルの場合である。

【0036】次に、ステータス関数の優先度の算出方法について、ここでは2例を挙げて説明する。なお、ステータス関数の優先度の算出方法は、ここに挙げるものだけに限るものではない。

【0037】第1の算出方法としては、各ステータス関数は、それに対応した入力バッファの占有量を調べ、その占有量をある規則に従って優先度に変換する。ある規

則とは、例えば、占有量とあらかじめ予想される入力データのレートから、オーバーフローするまでの時間を推測し、その時間が短いほど優先度が高くなり、その時間が長いほど優先度が低くなるようなテーブルにより変換する。

【0038】第2の算出方法としては、各ステータス関数は、それに対応したタスクが処理を完了しているデータのタイム・スタンプを調べ、その時刻と実際の時刻との差を優先度に変換して返す。この場合は、入力データIDATA₁～IDATA_nがすべてタイム・スタンプを持ったデータであることが条件である。また、処理装置1には現在時刻を表す時計またはカウンタが具備されているものとする。この例では、実時間に対する遅延が最も大きいものに最も高い優先度が与えられる。

【0039】次に、図3を参照して、制御プログラム2のマルチタスク制御処理(処理B)について説明する。

【0040】図3に示す処理Bは、次の処理対象としてのタスクを決定する際の判断基準としてのコマンド存在が、他の判断基準より重要度が低い場合の方法である。すなわち、まず、制御プログラム2は、各タスクのコマンドの有無を無視して、上記他の判断基準に基づき各タスクの優先度を算出する(ステップS11)。この優先度の算出方法は上記同様であってもよい。この算出した各タスクの優先度のうち、最も優先度の高いタスクが複数存在する場合にのみ(ステップS12)、この複数のタスクについて、それぞれのコマンドボックスにコマンドが書き込まれているか否かをチェックし、コマンドを受けているタスクを次に処理すべきタスク(k)として選択する(ステップS13)。

【0041】なお、ステップS13で、優先度の高いタスクのうち、コマンドを受けているタスクも複数存在するときは、そのうちのコマンドの数が多いものを優先にタスク(k)として選択してもよい。また、ステップS12で、最も優先度の高いタスクが1つのみときは、そのまま、ステップS14へ進み、当該タスクをタスク(k)とする。

【0042】次に、図4を参照して、制御プログラム2のマルチタスク制御処理(処理C)について説明する。

【0043】図4に示す処理Cは、コマンドの存在を各タスクの優先度自体にポイントとして加算して、コマンドの存在を反映した優先度を用いる方法である。すなわち、コマンドボックスにコマンドが書き込まれている場合には、コマンド1につき、優先度に相当する所定ポイントを予め決めておく。制御プログラム2は、まず、各タスクのコマンドの有無を無視して、各タスクの優先度を算出する(ステップS21)。この優先度の算出方法は上記同様であってもよい。そして、各タスクのそれぞれのコマンドボックスにコマンドが書き込まれているか否かを確認し、コマンドを受けているタスクがあれば、そのタスクの上記優先度に加え書き込まれたコマンド

の数に応じて、上記所定ポイントを加算する(ステップS22)。あるいは、コマンドが1つ以上書き込まれている場合には、その数に無関係に所定ポイントを上記優先度に加算するようにしてもよい。

【0044】この優先度が最も高いタスクを次に実行すべきタスク(k)として選択する(ステップS23)。

【0045】なお、上記処理Cでは、コマンドの種類に関係なく、コマンドボックスに書き込まれている1つのコマンドにつき、一律に所定ポイントを加算するものであったが、コマンドの種類に応じて、異なるポイントを上記優先度に加算するようにしてもよい。すなわち、コマンドの種類によっては、その緊急度が異なる場合がある。例えば、リアルタイム性が求められるコマンド(例えば、出力データに直接影響するようなコマンド)は、早く応答が必要があるが、そうでないものは、緊急性が要求されずCPUが空いているときに実行しよといいうものである。この場合、コマンドの種類に対応して、そのコマンドの緊急度に応じたポイントを予め定め、例えば、図5に示すようなコマンドテーブルを予め設定しておく。制御プログラム2は、図4のステップS22で上記優先度に加算する際には、そのコマンドの種類に対応したポイント値を図5に示したコマンドテーブルから読みとって、それを上記優先度に加算する。コマンドボックスに複数のコマンドが書き込まれている場合は、そのそれぞれのコマンドの種類に対応したポイント値をコマンドテーブルから読みとって、それを上記優先度に加算すればよい。

【0046】以上説明したように、上記実施形態によれば、制御プログラム2は、タスクがコマンドを受け付けていることを、次の処理対象のタスクを決定する際の複数の判断基準のうちの1つとし、コマンドを受け付けたタスクを優先的に次の処理対象のタスクとして決定することにより(例えば、コマンドを受け付けたタスクのそれぞれについて、他の判断基準に基づき優先度を算出し、この算出された優先度に基づき次の処理対象のタスクとして決定する。あるいは、コマンドを受け付けているタスクの有無を無視して、全てのタスクについて、他の判断基準に基づきそれぞれ優先度を算出し、この算出された優先度が最も高いものが複数あるときのみ、その複数のタスクのうち、コマンドを受け付けているタスクを次の処理対象のタスクとして決定する。)、コマンドの応答性を損なわずにマルチタスク処理を可能にする。例えば、ストリーミング処理におけるマルチタスク処理中に外部からコマンドが入力されたときでも、コマンドの応答性を損なわずにマルチタスク処理を可能にする。

【0047】また、タスクが受け付けているコマンドの有無およびコマンドの数およびコマンドの種類のうち少なくとも1つを優先度を算出するための複数の判断基準のうちの1つとして各タスクのそれぞれの優先度を算出して、この優先度に基づき次の処理対象のタスクを決

定することにより、コマンドの応答性を損なわずにマルチタスク処理を可能にする。例えば、ストリーミング処理におけるマルチタスク処理中に外部からコマンドが入力されたときでも、コマンドの応答性を損なわずにマルチタスク処理を可能にする。

【0048】さらに、次の処理対象のタスクを決定する際には、上記他の判断基準の1つとして、各タスクの処理の入出力系統の現状を考慮することにより、複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するデータ処理をマルチタスク処理にて実行することが可能となり、例えば、ストリーミングの際に、処理が所定の時刻に間に合うように行われ、入力バッファがオーバーフローしたり、出力がとぎれたりすることがなくなる。

【0049】なお、本発明は、上記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。さらに、上記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより、種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の全てを述べた課題(の少なくとも1つ)が解決でき、発明の効果の全てを述べられている効果(の少なくとも1つ)が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【0050】また、本発明の実施の形態に記載した本発明の手法は、コンピュータに実行させることのできるプログラムとして、磁気ディスク(フロッピー(登録商標)ディスク、ハードディスクなど)、光ディスク(CD-ROM、DVDなど)、半導体メモリなどの記録媒体に格納して頒布することもできる。

【0051】

【発明の効果】以上説明したように、本発明によれば、コマンドの応答性を損なわずにマルチタスク処理を可能にする。

【図面の簡単な説明】

【図1】本発明の実施形態に係る処理装置の要部の構成を概念的に示した図。

【図2】制御プログラムのマルチタスク制御処理(処理A)動作を説明するためのフローチャート。

【図3】制御プログラムのマルチタスク制御処理(処理B)動作を説明するためのフローチャート。

【図4】制御プログラムのマルチタスク制御処理(処理C)動作を説明するためのフローチャート。

【図5】コマンドテーブルの一例を示した図。

【符号の説明】

IDATA_1〜IDATA_n…入力データ

IBF_1〜IBF_n…入力バッファ

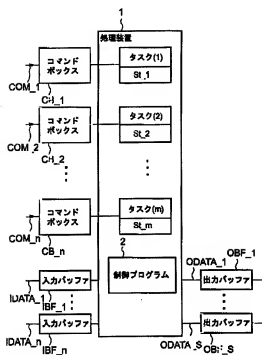
St_1〜St_m…ステータス関数

ODATA_1〜ODATA_s…出力データ

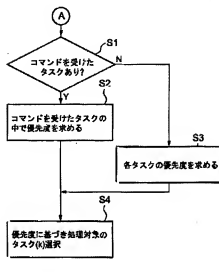
OBF_1~OBF_s...出力バッファ
COM_1~COM_n...コマンド

CB_1~CB_n...コマンドボックス

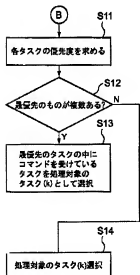
【図1】



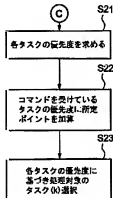
【図2】



【図3】



【図4】



【図5】

コマンドテーブル

コマンドの種類	ポイント
コマンドa	x1
コマンドb	x2
コマンドc	x3
...	...